



(12) **United States Patent**
Gerra et al.

(10) **Patent No.:** US 6,630,942 B2
(45) **Date of Patent:** *Oct. 7, 2003

- (54) METHODS AND APPARATUS FOR ACCESSING INFORMATION FROM MULTIPLE REMOTE SOURCES

- (75) Inventors: **Michael Gerra**, Dallas, TX (US);
Joseph Robert Offutt, Jr., Grapevine,
TX (US)

- (73) Assignee: Sabre Inc., Southlake, TX (US)

- (*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

- (21) Appl. No.: 09/847,337

- (22) Filed: **May 3, 2001**

- (65) **Prior Publication Data**

US 2001/0022591 A1 Sep. 20, 2001

Related U.S. Application Data

- (63) Continuation of application No. 09/031,674, filed on Feb. 27, 1998, now Pat. No. 6,229,534.

- (51) **Int. Cl.⁷** **G09G 5/00**

- (52) U.S. Cl. **345/744**; 345/733; 345/804;
345/854; 709/203; 709/219

- (58) **Field of Search** 345/744, 733,
345/804, 854, 840; 709/203, 217, 219

- (56)
- References Cited**

U.S. PATENT DOCUMENTS

- | | | | |
|-------------|---------|--------------------|---------|
| 4,788,643 A | 11/1988 | Trippe et al. | 705/6 |
| 5,311,425 A | 5/1994 | Inada | 705/6 |
| 5,649,186 A | 7/1997 | Ferguson | 395/610 |
| 5,754,830 A | 5/1998 | Butts et al. | 395/500 |

- | | | | |
|---------------|---------|----------------------------|---------|
| 5,768,578 A | 6/1998 | Kirk et al. | 395/611 |
| 5,796,393 A | 8/1998 | MacNaughton et al. | 345/733 |
| 5,809,248 A | 9/1998 | Vidovic | 709/219 |
| 5,838,906 A | 11/1998 | Doyle et al. | 709/202 |
| 5,864,827 A | 1/1999 | Wilson | 705/35 |
| 5,867,153 A | 2/1999 | Grandcolas et al. | 345/700 |
| 5,893,093 A | 4/1999 | Wills | 707/5 |
| 5,940,075 A | 8/1999 | Mutschler, III et al. | 345/744 |
| 5,978,834 A | 11/1999 | Simonoff et al. | 709/203 |
| 6,028,602 A * | 2/2000 | Weidenfeller et al. | 345/781 |
| 6,229,534 B1 | 5/2001 | Gerra et al. | 395/611 |

FOREIGN PATENT DOCUMENTS

- | | | |
|----|--------------|--------|
| EP | 0 660 251 A3 | 6/1995 |
| EP | 0 660 251 A2 | 6/1995 |

OTHER PUBLICATIONS

PCT Search Report for International Application No. PCT/
US 99/04466.

Tom Tessier, "Sharing Data Between Web Page Frames Using JavaScript," *Programmer's Workbench*, Dr. Dobb's Journal, May 1996, pp. 72, 74, 75, 86, 87.

* cited by examiner

Primary Examiner—John Cabeca

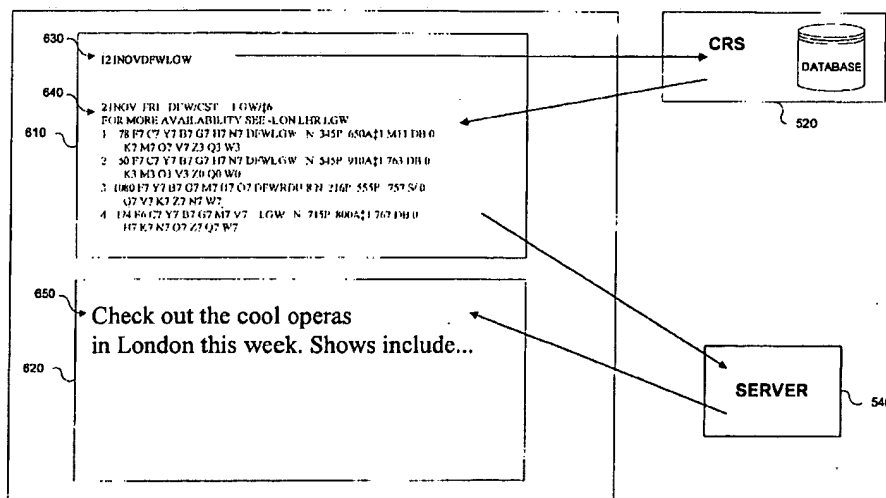
Assistant Examiner—Cuong T. Thai

(74) *Attorney, Agent, or Firm*—Alston & Bird LLP

(57) **ABSTRACT**

In accordance with the present invention a process is provided for interfacing a legacy application and internet-based application and for displaying information from each application in a frame of a multi-frame browser on a client workstation. Further, the interface system enables users to transact business with the legacy application based on information retrieved from the internet-based application, and to retrieve information from the internet-based application based on information retrieved from the legacy application.

19 Claims, 8 Drawing Sheets



no server for
providing
legacy host screen

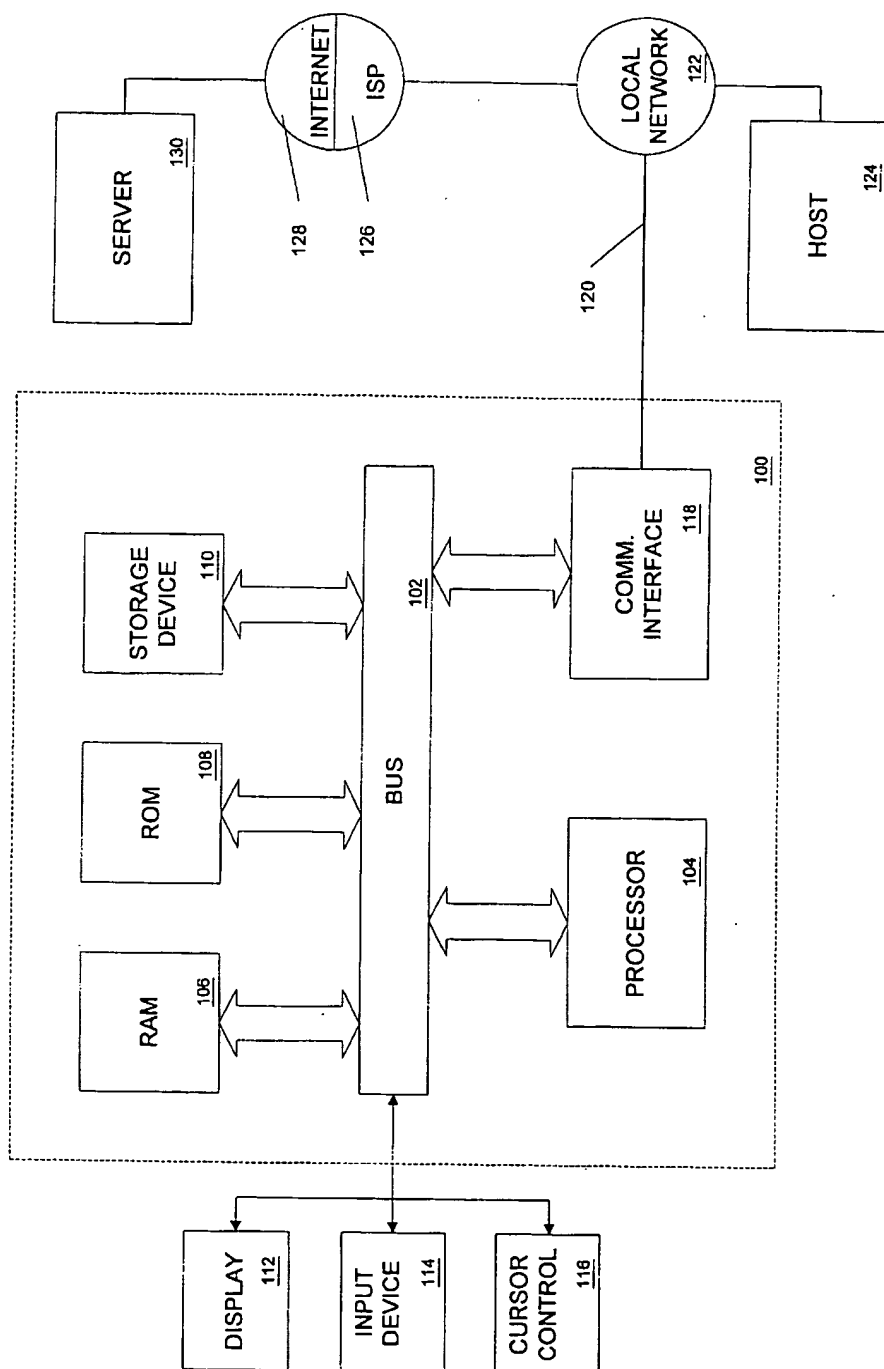
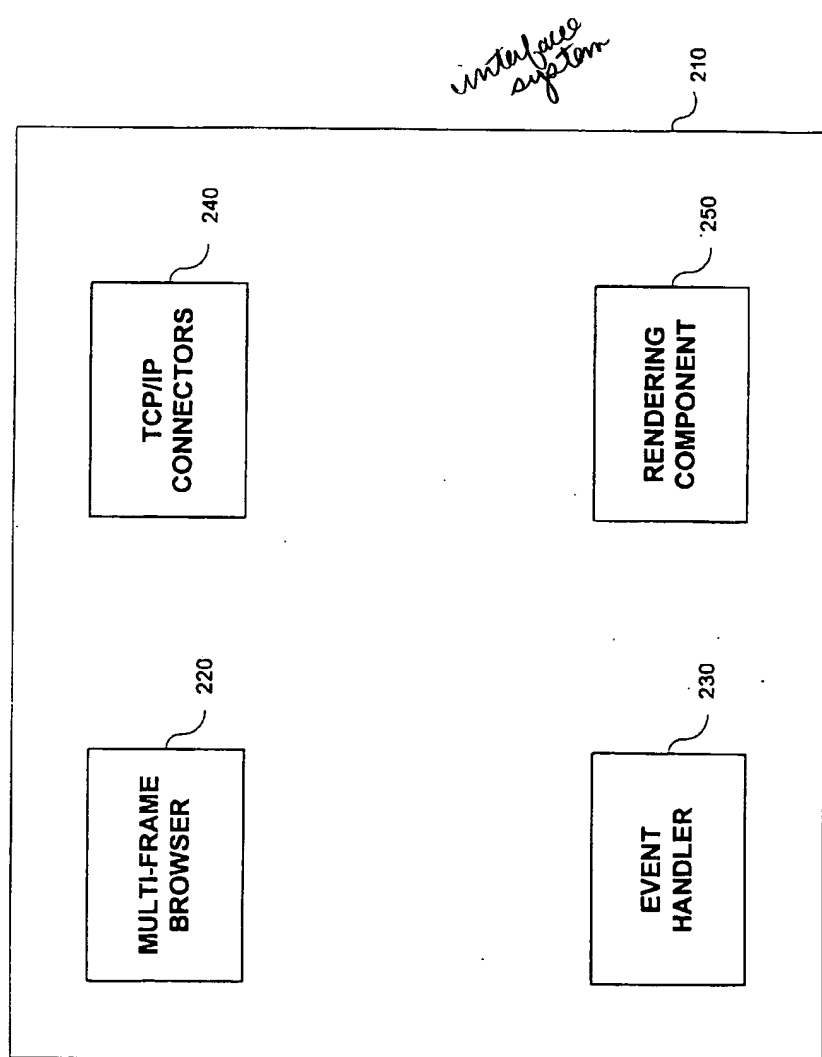


FIG. 1

**FIG. 2**

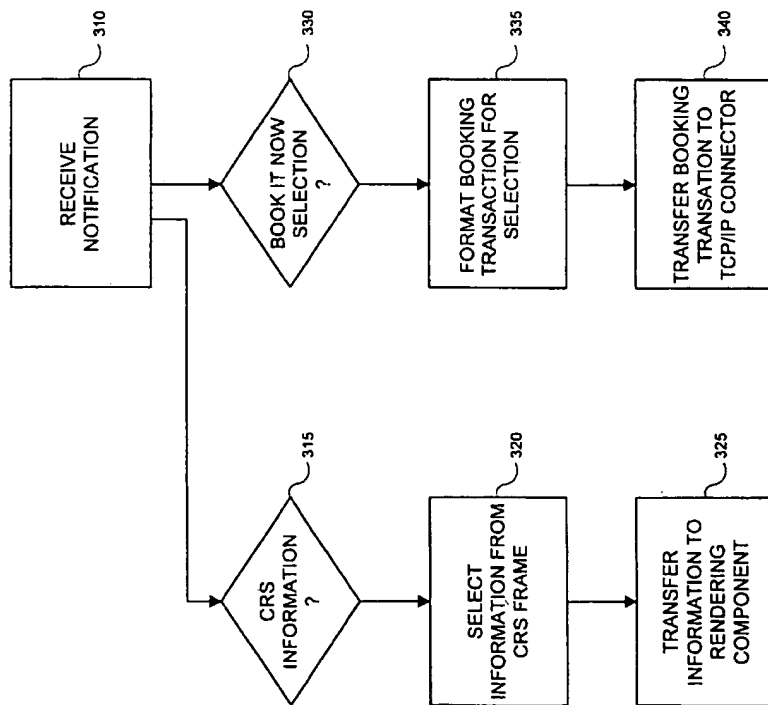
EVENT
HANDLER

FIG. 3

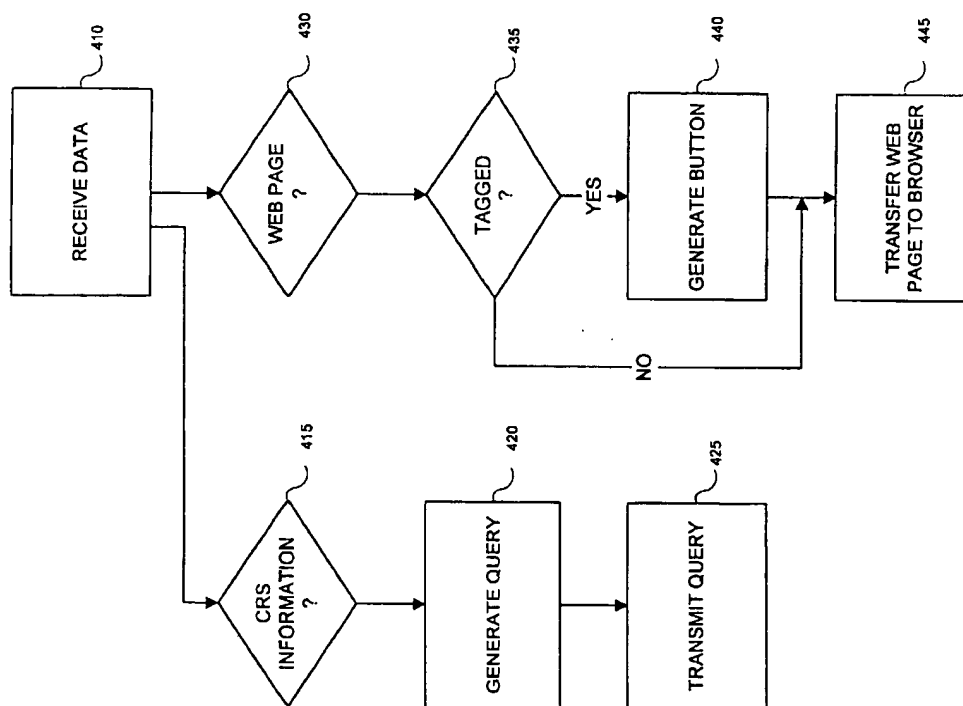
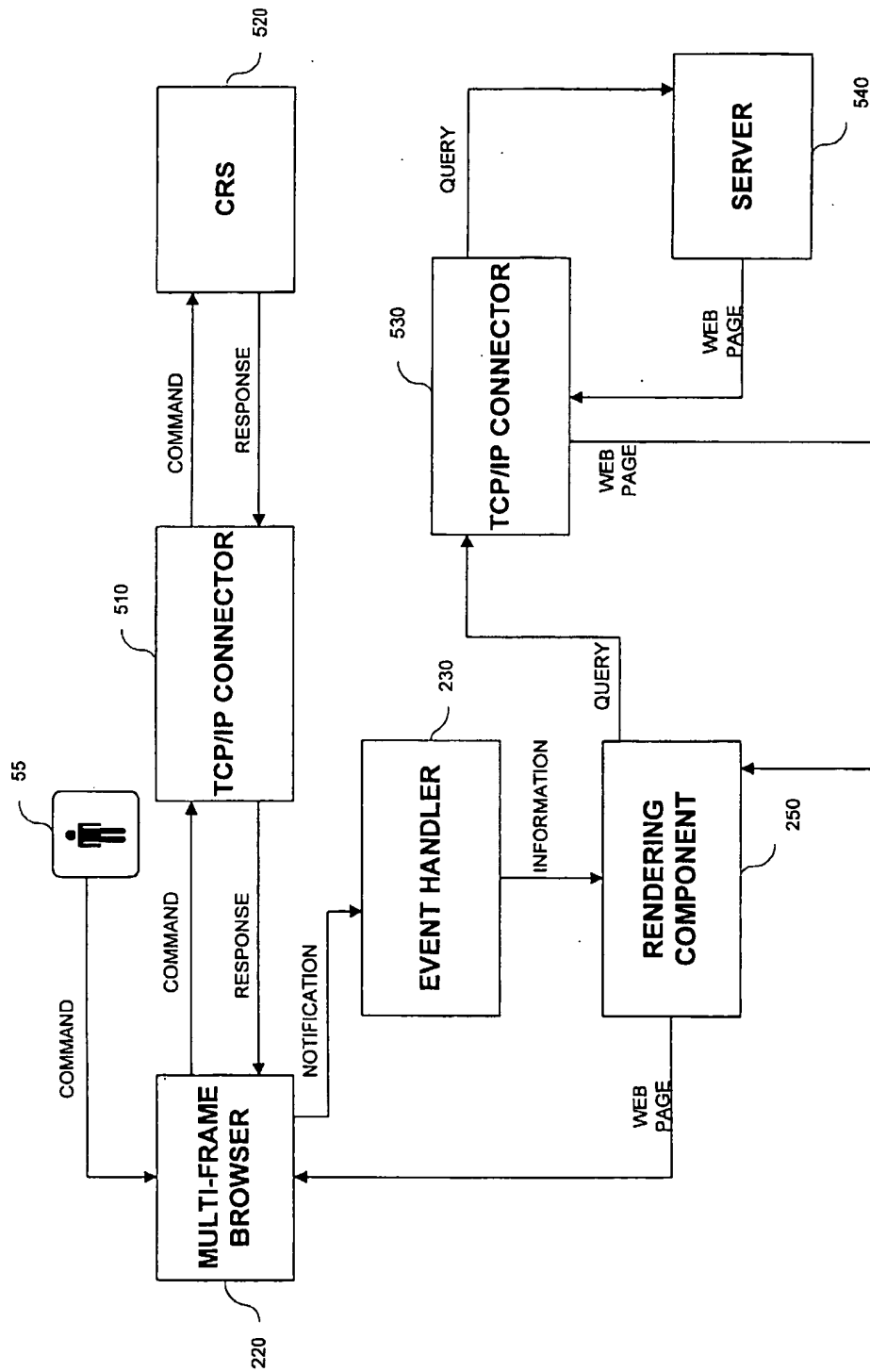
RENDERING
COMPONENT

FIG. 4

**FIG. 5**

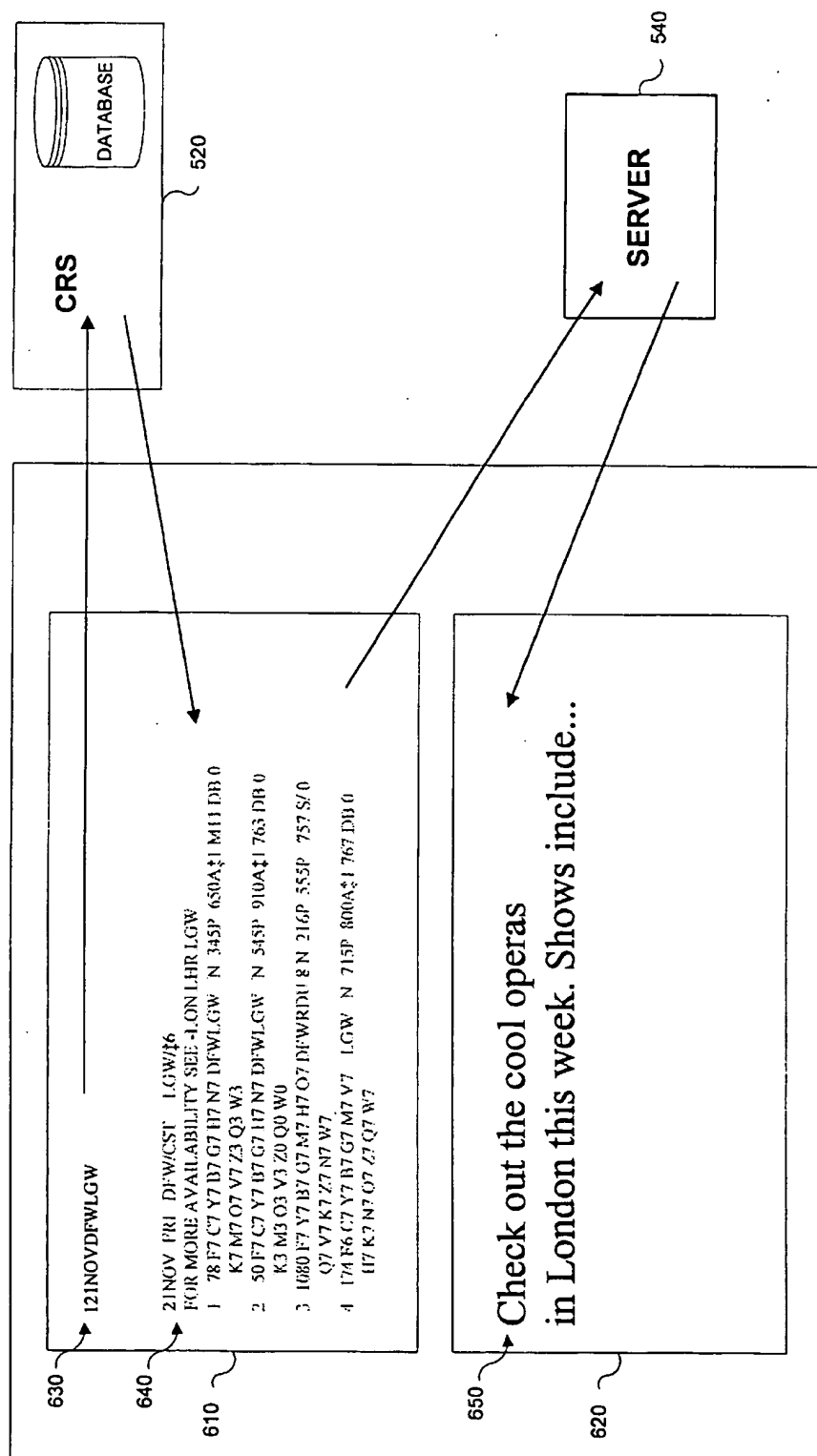


FIG. 6

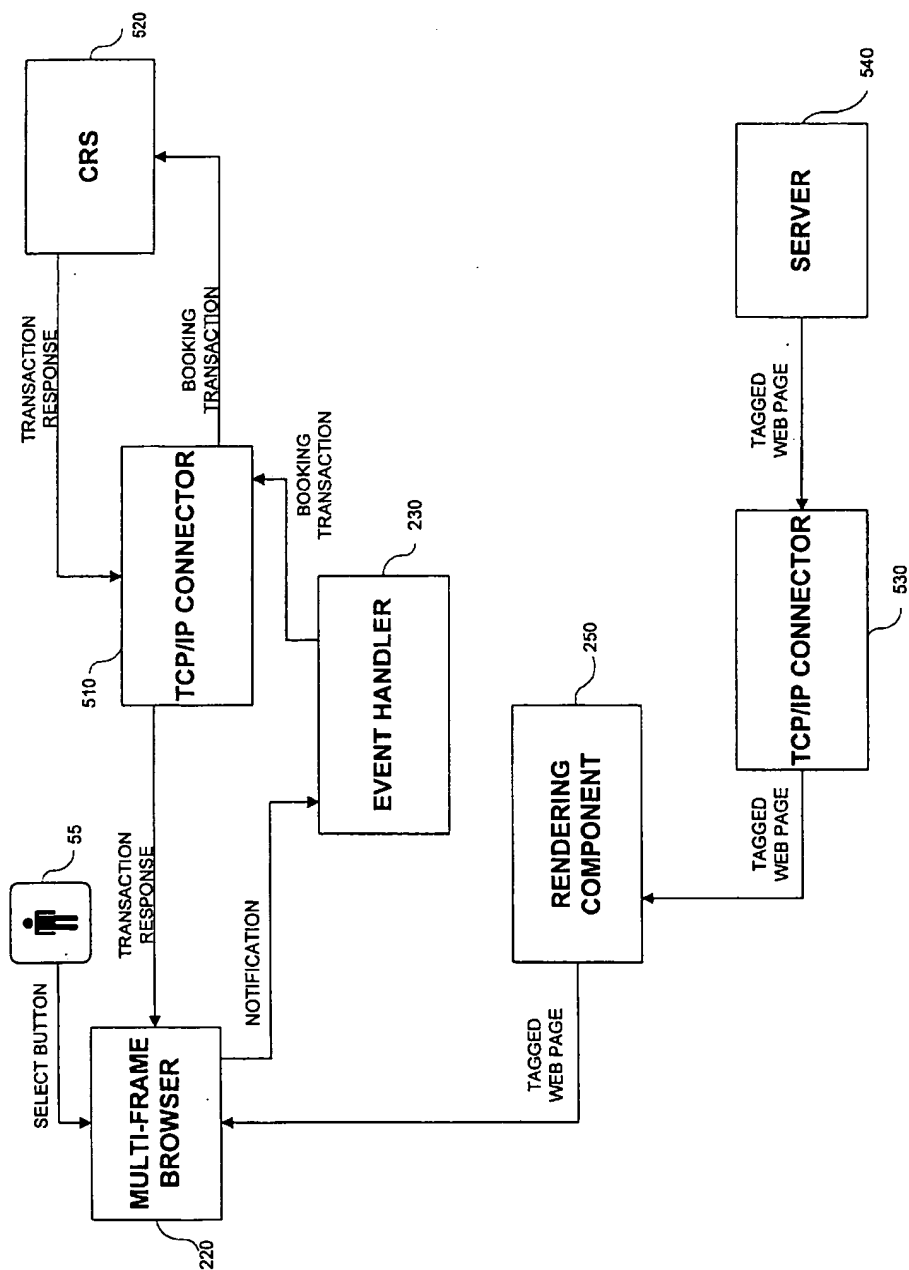


FIG. 7

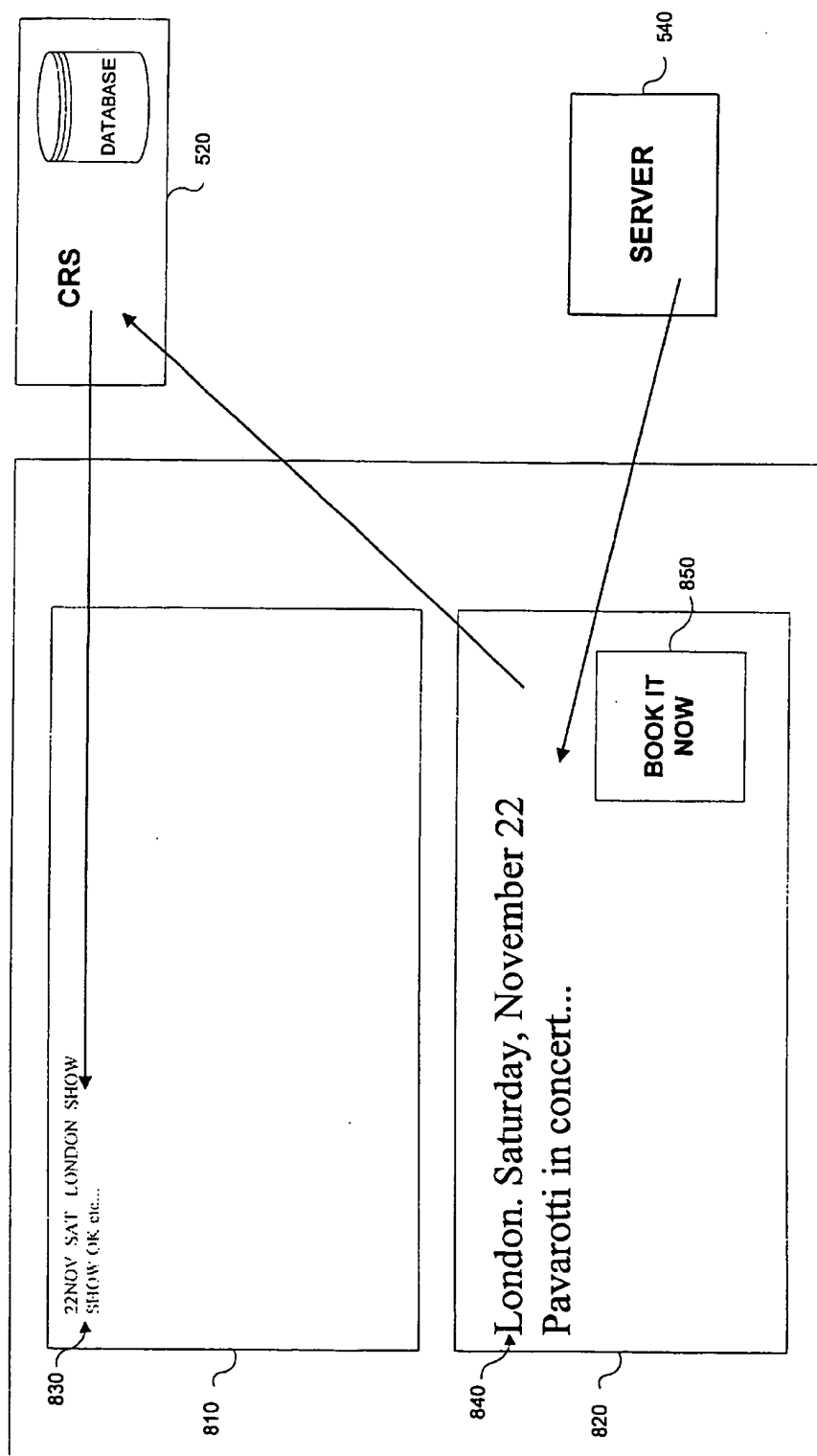


FIG. 8

1

METHODS AND APPARATUS FOR ACCESSING INFORMATION FROM MULTIPLE REMOTE SOURCES

This is a continuation of application Ser. No. 09/031,674, filed Feb. 27, 1998 U.S. Pat. No. 6,229,534, which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

A. Field of the Invention

This invention generally relates to interfaces for computer systems and, more particularly, to an interface for viewing in separate frames information from at least two remote sources and communicating information between the frames to enhance the user's access to information from the sources.

B. Description of the Related Art

The Internet, fueled by the phenomenal popularity of the World Wide Web (WWW), has exhibited exponential growth over the past few years. In the case of the WWW, the ease of self-publication has helped generate an estimated 50-120 million documents on a broad range of subjects.

To access all this information, users need only standard computer equipment, such as a personal computer with a display and modem, and an Internet connection. Several types of Internet connections are available, including connections through Internet Service Providers (ISPs). To use an Internet connection from an ISP, for example, the user dials into a computer at the ISP's facility using the modem and a standard telephone line. The ISP's computer in turn provides the user with access to the Internet.

Through this Internet connection, the user accesses information on the web using a computer program called a "web browser," such as the Netscape Navigator™ from Netscape Communications Corporation. To accomplish this, the user gives the web browser a Uniform Resource Locator (URL) for an object on the Internet, for example, a document containing information of interest. The document is referred to as a "web page," and the information contained in the web page is called "content." Web pages often refer to other web pages using "hypertext link" or "hyperlinks" that include words or phrases representing the other pages in a form that gives the browser a URL for the corresponding web page when a user selects a hyperlink. Hyperlinks are made possible by building web pages using the Hypertext Markup Language (HTML).

The URL identifies a specific computer on the Internet, called a "Web Server," and, more particularly, the location of a web page located on the web Server. The web browser retrieves the web page and displays it for the user.

The Internet thus provides users access to a wide variety of information. For example, users can use the Internet to locate information on current and upcoming events in cities and communities throughout the world.

In contrast, companies offering access to specific types of information have used and continue to use stand-alone, client-server, or client-to-host applications that deliver specific or proprietary functionality and related information to a client workstation. For examples in the travel industry the SABRE® System provides users such as travel agents with access to numerous databases of travel information for the purpose of making customer reservations with airlines, hotels, car rental agencies, etc. SABRE® is a registered trademark of TSGL Holdings, Inc.

The travel industry is dominated by such legacy systems because long ago certain travel industry groups, notably the

2

United States-based airline industry, settled on several computer-based reservation systems (CRSs) for managing flight reservation information based on each traveler's desired origination-to-destination journey. CRSs generally arrange, organize, and store this data in a format generally corresponding to the dominant travel routing patterns, in which a carrier accepts a passenger at a first, origination location and discharges the passenger at a terminal or destination location. The data is generally stored in several databases within each reservation system: schedules, tariffs, and travel rules.

Booking a flight through a CRS involves creating a computer record called the Passenger Name Record (PNR) of data from each of these databases. As flights are booked for different carriers, the system sends a message with the flight information to the airline's computers. The PNR is used to generate tickets and itineraries.

Current legacy applications, such as CRSs, and newer HTML-based Internet applications deliver discrete and separate information streams to a client workstation. While a user may be able to use both applications at the same time, the user is not able to share events or information across these applications because the applications function totally independently of each other. For example, a travel agent can display rich destination content for New York City from a web site with the help of a web browser. Then, by switching to a different window on the same workstation, the agent can use a CRS such as the SABRE System to display a list of airline flights to New York. However, the two information displays are in no way linked, either physically or logically. This requires the travel agent to be familiar with two different systems, different data sources, different windows, and different transactions.

There is therefore a need for a system that bridges the gap between these two environments by interfacing at least two applications, such as a legacy application and HTML-based Internet application, and for displaying information from each application in a frame of a multi-frame browser on a client workstation.

SUMMARY OF THE INVENTION

Accordingly, systems and methods consistent with the present invention substantially obviate one or more of the problems due to limitations, shortcomings, and disadvantages of the related art by bridging legacy applications and internet-based applications at the client workstation. In accordance with the present invention a process is provided for interfacing at least two applications, such as a legacy application and internet-based application, and for displaying information from each application in a frame of a multi-frame browser on a client workstation. Further, the interface system enables users to transact business with the first application based on information retrieved from the second application, and to retrieve information from the second application based on information retrieved from the first application.

In accordance with the present invention, as embodied and broadly described herein, a method for accessing information from multiple remote resources using a workstation having a display and a connector for connecting to the remote systems comprises (i) receiving a record from a first remote resource; (ii) displaying data associated with the record in a first frame on the display; (iii) transmitting a request corresponding to the displayed data to a second remote resource; and (iv) displaying data associated with the transmitted request in a second frame on the display. The

3

transmitting step involves parsing the record for selected information, and generating the request using the selected information. The step of displaying data associated with the transmitted request in a second frame involves receiving a response to the transmitted request, and building a page for display in the second frame. The step of transmitting a request may also include identifying the second remote system based on the displayed data in the first frame. The step of displaying data associated with the transmitted request in a second frame on the display includes determining whether the data associated with the transmitted requests contains a tag, and displaying a button corresponding to the tag within the second frame.

Both the foregoing general description and the following detailed description are exemplary and explanatory only, and merely provide further explanation of the claimed invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the drawings,

FIG. 1 is a schematic block diagram of a computer architecture suitable for use with the present invention;

FIG. 2 is a block diagram of components of an interface system consistent with the present invention;

FIG. 3 is a flow diagram of the steps performed by the event handler of the interface system consistent with the present invention;

FIG. 4 is a flow diagram of the steps performed by the rendering component of the interface system consistent with the present invention;

FIG. 5 is a block diagram used to explain a flow of information using the interface system consistent with the present invention;

FIG. 6 is a view of the output displayed in connection with the information flow of FIG. 5;

FIG. 7 is a block diagram used to explain another flow of information using the interface system consistent with the present invention; and

FIG. 8 is a view of the output displayed in connection with the information flow of FIG. 7.

DETAILED DESCRIPTION

Reference will now be made in detail to an implementation of the present invention as illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings and the following description to refer to the same or like parts.

Overview

Systems consistent with the present invention include several components: (1) a browser with a multi-frame viewer with each frame displaying information from a different source; (2) an event handler that observes events in the frames; and (3) a rendering component that moves information in both directions between frames. For example, the rendering component captures an event observed inside one frame corresponding to a first application (such as a legacy application), parses data related to the event, and delivers information from the frame into a format for delivery to a second application (such as an HTML-based

4

application). The rendering component may also parse user-defined functions embedded in the data stream from the second application delivered to a corresponding frame, and convert those functions into a format for delivery to the first application. Additional components provide the necessary connections between the client workstation and the remote resources.

Computer Architecture

The present invention may be implemented by computers organized in a conventional distributed processing system architecture. The architecture for and procedures to implement this invention, however, are not conventional, because they bridge multiple remote information sources, such as a legacy computer application (e.g., CRS) and an HTML-based Internet application, at the client workstation.

FIG. 1 is a block diagram that illustrates a computer system 100 upon which an embodiment of the invention may be implemented. Computer system 100 includes a bus 102 or other communication mechanism for communicating information, and a processor 104 coupled with bus 102 for processing information. Computer system 100 also includes a main memory, such as a random access memory (RAM) 106 or other dynamic storage device, coupled to bus 102 for storing information and instructions to be executed by processor 104. RAM 106 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 104. Computer system 100 further includes a read only memory (ROM) 108 or other static storage device coupled to bus 102 for storing static information and instructions for processor 104. A storage device 110, such as a magnetic disk or optical disk, is provided and coupled to bus 102 for storing information and instructions.

Computer system 100 may be coupled via bus 102 to a display 112, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 114, including alphanumeric and other keys, is coupled to bus 102 for communicating information and command selections to processor 104. Another type of user input device is cursor control 116, such as a mouse, a trackball or cursor direction keys for communicating direction information and command selections to processor 104 and for controlling cursor movement on display 112. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

The invention is related to the use of computer system 100 for accessing information from multiple remote resources. According to one implementation of the invention, information from the multiple remote resources is provided by computer system 100 in response to processor 104 executing one or more sequences of one or more instructions contained in main memory 106. Such instructions may be read into main memory 106 from another computer-readable medium, such as storage device 110. Execution of the sequences of instructions contained in main memory 106 causes processor 104 to perform the process steps described herein. In an alternative implementation, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus implementations of the invention are not limited to any specific combination of hardware circuitry and software.

The term "computer-readable medium" as used herein refers to any media that participates in providing instructions to processor 104 for execution. Such a medium may take

5

many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 110. Volatile media includes dynamic memory, such as main memory 106. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 102. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, papertape, any other physical medium with patterns of holes, a RAM, PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 104 for execution. For example, the instructions may initially be carried on magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 100 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector coupled to bus 102 can receive the data carried in the infra-red signal and place the data on bus 102. Bus 102 carries the data to main memory 106, from which processor 104 retrieves and executes the instructions. The instructions received by main memory 106 may optionally be stored on storage device 110 either before or after execution by processor 104.

Computer system 100 also includes a communication interface 118 coupled to bus 102. Communication interface 118 provides a two-way data communication coupling to a network link 120 that is connected to local network 122. For example, communication interface 118 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 118 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 118 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

Network link 120 typically provides data communication through one or more networks to other data devices. For example, network link 120 may provide a connection through local network 122 to a host computer 124 and/or to data equipment operated by an Internet Service Provider (ISP) 126. ISP 126 in turn provides data communication services through the Internet 128. Local network 122 and Internet 128 both use electric, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 120 and through communication interface 118, which carry the digital data to and from computer system 100, are exemplary forms of carrier waves transporting the information.

Computer system 100 can send messages and receive data, including program code, through the network(s), network link 120 and communication interface 118. In the Internet example, a server 130 might transmit a requested

6

code for an application program through Internet 128, ISP 126, local network 122 and communication interface 118. In accordance with the present invention, one such downloaded application provides a bridge between two remote information sources, as described herein. The received code may be executed by processor 104 as it is received, and/or stored in storage device 110, or other non-volatile storage for later execution. In this manner, computer system 100 may obtain application code in the form of a carrier wave.

Although computer system 100 is shown in FIG. 1 as being connectable to one server, 130, those skilled in the art will recognize that computer system 100 may establish connections to multiple servers on Internet 128. Each such server includes an HTML-based Internet application, which may provide information to computer system 100 upon request in a manner consistent with the present invention. Additionally, for purposes of this description host 124 comprises a legacy computer application such as a CRS, although other legacy applications may be used to implement the present invention.

Interface (Split-Screen) System

FIG. 2 is a block diagram of components of an interface system 210 consistent with the present invention. A primary purpose of interface system 210 is to present multiple frames of information on display 112 by, in part, using information displayed in one frame to retrieve and display information in another frame. Thus, interface system 210 separates a display screen into multiple (at least two) frames to display information from multiple sources. Although this description is directed to an implementation having a legacy application for one source and a legacy application(s) and/or HTML-based application(s) for other sources, those skilled in the art will recognize that any combination of at least two sources may be used to implement the present invention.

Interface system 210 is a software application preferably written in the Java™ programming language, although other programming languages may be used. The Java programming language is an object-oriented programming language that is described, for example, in a text entitled "The Java Language Specification" by James Gosling, Bill Joy, and Guy Steele, Addison-Wesley, 1996. Programs written in the Java language are typically compiled to a universal executable format, using a "bytecode instruction set," which can be executed on any platform supporting the Java virtual machine (VM). The Java VM is described, for example, in a text entitled "The Java Virtual Machine Specification," by Tim Lindholm and Frank Yellin, Addison Wesley, 1996. Thus, interface system 210 can reside on any client workstation, such as computer system 100, that supports a Java VM.

Interface system 210 is comprised of a multi-frame browser 220, an event handler 230, TCP/IP connectors 240, and a rendering component 250. Multi-frame browser 220 is comprised of a customizable HTML browser for generating multiple frames for display, each frame designated to display information from at least one source. A suitable browser for this purpose is the "ICE Browser" from ICESoft A/S of Bergen, Norway (<http://www.icesoft.no/>), because it is extensible, meaning that it can be configured to operate with the other components of system 210 such as handler 230 and rendering component 250.

Event handler 230 and rendering component 250 are extensions of multi-frame browser 220. These extensions observe events in each frame and, when appropriate, retrieve and display information in another frame.

TCP/IP connectors 240 connect interface system 210 operating on computer system 100 to remote sources, such as a legacy application operating on host 124, and an HTML-based application operating on server 130. "TCP/IP" is used in this description to provide an example of protocols capable of being used with systems consistent with the present invention. This does not in any way limit the invention to TCP/IP connections, other types of connection such as X.25 and SNA (System Network Architecture) may also be used within the scope of this invention.

A TCP/IP connector to a legacy application typically uses a set of low level communications functions referred to as an application programming interface (API). One such API has been developed by The SABRE Group for general client workstation access to the SABRE host system, which is an exemplary legacy CRS application. The Java Common SABRE API (JSAPI) is in the form of Java classes. The JSAPI is based on a three layer architecture in which each layer handles a specific function: (1) a device layer (2) a connection layer, and (3) a services layer.

The first two layers are common to all TCP/IP connections. The device layer handles all the session management functions, which are described in various texts on TCP/IP connections, including for example, W. Richard Stevens, TCP/IP Illustrated, Vol. 1, Addison-Wesley, 1994. The connection layer handles all of the connection specific details, which are also described in the aforementioned text by W. Richard Stevens. However, the services layer is comprised of host specific objects, called Service Providers, which are loaded when required to map SABRE LnLaTa terminal addresses to TCP/IP connections. LnLaTa, which stands for Line Interchange Address Terminal Address, is the unique 24-bit terminal address that identifies a user of system resources on the SABRE host, such as a terminal, workstation, or printer. The user typically connects through a Local Area Network (LAN) to an IP router that in turn connects the LAN to a secure dedicated network.

When a JSAPI connection is created on a client workstation, a session string or reference is used to query a registry or table, which contains system configuration information used to configure and control applications like system 210, to locate the desired Service Provider. Service Providers are program modules that allow applications to communicate with different end systems such as an Open Front End Processor (OFEP), or some other gateway, connected by an IP router to the dedicated network. Service Providers are typically dynamically loaded based on configuration information in the registry.

The OFEP is a TCP/IP interface to a Lineside MicroVax Front End (MFE) processor, which interfaces through another network with a Hostside MFE processor connected to the host processor of the SABRE host system. The MicroVax is a computer manufactured by Digital Equipment Corporation. The OFEP communicates with the Lineside MFE using the DECnet protocol while the Lineside MFE communicates with the Hostside MFE using the Ethernet protocol. The Hostside MFE in turn uses a channel protocol to communicate with the host computer. The OFEP is therefore responsible for actual data transfer with the SABRE host over the TCP/IP protocol, and manages incoming and outgoing I/O requests.

If the user does not specify a specific Service Provider by name, the first Service Provider enumerated in the registry is loaded. One of these Service Providers is the SABRE Front End Processor that provides connectivity to the SABRE host's various mainframe computers. The SABRE

FEP also provides other value-added services, such as switchable printer support and connectivity to other external legacy systems.

In contrast, the TCP/IP connector to the HTML-based application uses the standard internet HTTP (Hyper Text Transfer Protocol) or similar protocols, which are explained in detail in the aforementioned text by W. Richard Stevens.

Event Handler

The process steps of event handler 230 will now be described with reference to FIG. 3. For purposes of this description, multi-frame browser 220 displays two frames, a first frame corresponding to a legacy application such as a CRS and a second frame corresponding to an HTML-based application.

In general, event handler 230 reacts to changes in the state of each frame of browser 220. When the state of a frame changes, event handler 230 receives a notification of that change (step 310). For purposes of this description, there are two state changes that cause event handler 230 to receive a notification: (1) display of information in the frame corresponding to the legacy application (step 315); and (2) user selection of a "book it now" (BIN) button (step 330).

If event handler 230 determines that a received notification indicates that browser 220 has displayed information in the frame corresponding to the legacy application (step 315), event handler 230 selects certain predetermined information from within that frame (step 320) and transfers the selected information to rendering component 250 (step 325). For example, when a legacy application frame displays schedule information, event handler 230 may select information identifying the passenger's destination to transfer to rendering component 250 so browser 220 can display in another frame information from a remote source, which is distinct from the legacy application, corresponding to the destination. This function of event handler 230 is explained further below in connection with FIGS. 5 and 6.

Under certain circumstances, browser 220 may display a BIN button in a frame. For example, the displayed information on a destination may include information on events scheduled to occur at the destination during a specific time period, e.g., after a date of arrival at the destination or prior to a date of departure from the destination as defined by the passenger's schedule. Browser 220 may display a BIN button for each event. When a user wishes to make a reservation to attend a selected event, or actually purchase a ticket to the event, the user merely selects the corresponding BIN button. When event handler 230 determines that a received notification indicates that a user has selected a BIN button displayed within a frame of browser 220 (step 330), event handler 230 formats a message to the legacy application, which message is referred to as a booking transaction, indicating the users selection (step 335). Event handler 230 transfers the booking transaction to the TCP/IP connector for the legacy application (step 340). This function of event handler 230 is explained further below in connection with FIGS. 7 and 8.

Rendering Component

FIG. 4 is a flow chart of the steps performed by the rendering component of interface system 210 consistent with the present invention. Rendering component 250 operates when it receives any one of two types of data (step 410): information comprised of a passenger's schedule, and a web page.

When rendering component 250 receives information on a passenger's schedule (step 415), it generates a query

corresponding to selected information from the information (step 420) and transmits the query to a TCP/IP connector (step 425). These two steps correspond to the event handler's selection of information from a display of schedule information (see step 315, FIG. 3). This function of rendering component 250 is explained further below in connection with FIGS. 5 and 6.

When rendering component 250 receives a web page from an HTML-based application on a remote server on the Internet (step 430), rendering component 250 first determines whether the page contains any tags (step 435). A tag, is a specific code in the web page indicating to the rendering component that a BIN button should be built for a corresponding piece of information in the web page. For example, when the legacy application has the facility to reserve or purchase tickets to an event, a web page with information on that event may be tagged so rendering component 250 generates a BIN button for the event. Thus, when a tag exists in the web page, rendering component 250 builds the corresponding BIN button (step 440). Then, or when no tag exists (step 435), rendering component 250 transfers the web page to browser 220 for display in the corresponding frame. This function of rendering component 250 is explained further below in connection with FIGS. 7 and 8.

Split Screen Interface

Reference will now be made to FIGS. 5 and 6 to explain how interface system 210 offers a scheme for communicating between legacy applications and HTML-based applications. As shown in FIG. 5, a user 55 inputs a command into a first frame (610, FIG. 6) of multi-frame browser 220. For example, the user may wish to retrieve certain schedule information from a legacy application such as CRS 520. As shown in FIG. 6, the user input command 630 is "121NOVDFWLGW". This command requests an itinerary ("I") for flights on the 21st of November ("21NOV") departing from Dallas-Fort Worth Airport ("DFW") that are destined for London Gatwick Airport ("LGW").

In this case, browser 220 uses a TCP/IP connector 510 for CRS 520 to send the user's command to CRS 520. In turn, CRS 520 provides a response to browser 220, via the same TCP/IP connector 510, which response includes the requested itinerary information 640. As shown in FIG. 6, response 640 indicates that there are 6 possible flights or combinations of flights on the 21st of November (which is a Friday) from DFW to LGW. Additionally, response 640 also indicates that more itineraries may be found using London Heathrow Airport (LHR). The first itinerary shows a number of codes i.e., "78 F7 C7 Y7 B7 G7 H7 N7" corresponding to, for example, the airfare and seat availability on a flight departing from DFW at 3:45 PM and arriving at LGW at 6:50 AM.

Event handler 230 receives a notification that response 640 has been displayed, and selects certain information from the response, e.g., the destination city, to retrieve additional information on scheduled events related to the selected information. The selected information is in turn provided to rendering component 250 that generates a query and transmits the query via another TCP/IP connector 530 to a server 540 on the Internet running an HTML-based application. Assuming this HTML-based application has information responsive to the query, server 540 responds with a web page that rendering component 250 receives via TCP/IP connector 530. The web page is provided to browser 220 for display in a frame different from the frame used to display the command and response. As shown in FIG. 6, a second frame 620 displays web page 650.

In this manner, interface system 210 effectively communicates information between a legacy application and an HTML-based application and displays information from both applications in a single multi-frame browser.

Book It Now

Continuing with FIGS. 7 and 8, the operation of interface system 210 in connection with the BIN button will now be explained. In this implementation, the retrieved web page includes a tag indicating a BIN button. Server 540 provides the tagged web page to rendering component 250 via TCP/IP connector 530. Rendering component 250 in turn provides the tagged web page with the generated BIN button to browser 220. As shown in FIG. 8, the web page 840 and BIN button 850 are displayed in frame 820 corresponding to CRS 520. Those skilled in the art will recognize that browser 220 may generate additional frames to display web pages and corresponding BIN buttons, if any, for additional servers. Alternatively, an aggregated web page and corresponding BIN buttons, if any, for all servers may be displayed in a single frame. In other words, although the figures show two frames, the first for the legacy application and the second for an HTML-based application, the present invention contemplates multiple frames, one for each legacy application and HTML-based application. Alternatively, a single frame may be used to display aggregated information from multiple legacy applications and a single frame may be used to display an aggregated web page from multiple HTML-based applications.

When user 55 selects a BIN button, event handler 230 receives an event notification, which causes handler 230 to generate a booking transaction to book a reservation for a ticket to the event corresponding to the BIN button. Handler 230 provides the transaction to TCP/IP connector 510 for CRS 520. After processing the transaction, CRS 520 transmits a transaction response that is provided to browser 220 via connector 510. This response is displayed in the frame dedicated to CRS 520.

As shown in FIG. 8, the user's selection of the BIN button for the Pavarotti concert in London on November 22nd, causes interface system 220 to generate a booking transaction to book a reservation for the concert by communicating the transaction to CRS 520, which in turn displays a transaction response 830 in the browser frame 810 dedicated to the application.

Conclusion

In accordance with the present invention an interface (split screen) system provides a two-way interface between two applications such as a legacy application and an HTML-based application running on the same client. A first frame of a browser supports a session with a first application, which may be a legacy application such as a CRS, and a second frame supports an internet browser session. The interface system passes relevant data between these frames based on context. For example, when a user requests a flight to London in the legacy CRS application frame the internet browser window responds by showing additional travel information specific to London. Similarly, when a user displays a bookable travel item in the internet browser frame, it can be booked by executing a corresponding CRS transaction, which then responds in the legacy CRS application frame.

The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the

11

invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes a BIN button used to book transactions identified by an HTML-based application via a legacy CRS application. Additionally, the interface system may provide means such as a button displayed in the frame dedicated to a legacy CRS application to retrieve additional information for display in the other frame corresponding to selected information displayed from the legacy CRS application. The scope of the invention is defined by the claims and their equivalents.

What is claimed is:

1. A user interface for accessing information from multiple remote sources comprising:
 - a first machine for displaying data in a first window; and
 - a second machine coupled to said first machine for:
 - extracting a portion of the displayed data from the first window;
 - composing a query from the extracted data;
 - transmitting the query to at least a third machine;
 - receiving results of the query from the third machine; and
 - displaying results of the query in a second window.
2. The user interface of claim 1 further comprising a display coupled to the first and second machines for displaying at least two windows.
3. The user interface of claim 2, wherein the first machine further comprises a processor for:
 - determining whether a page of the results contains a tag, and
 - displaying a button corresponding to the tag in the second window.
4. The user interface of claim 3, wherein the processor further comprises machine-readable instructions for:
 - determining when the button is activated;
 - extracting the results of the query from the second window when the button is activated;
 - composing a booking transaction from the results;
 - transmitting the booking transaction to the first machine; and
 - displaying a confirmation message in the first window.
5. The user interface of claim 1, wherein extracting a portion of the displayed data further comprises:
 - parsing the displayed data to locate text associated with an event, and
 - identifying text associated with the event.
6. The user interface of claim 1, wherein composing a query from the extracted data further comprises:
 - identifying at least a third machine based on the extracted data; and
 - reformatting the displayed data for delivery to said at least third machine.
7. The user interface of claim 1, further comprising a third machine for receiving the query.
8. The user interface of claim 7, wherein said third machine is also for storing data responsive to the query, and for providing the results of the query to the second machine.
9. A computer-implemented method for accessing information from multiple remote sources, comprising:
 - displaying data from a first machine in a first window;
 - extracting a portion of the displayed data from the first window;
 - composing a query from the extracted data;

12

- transmitting the query from a second machine to at least a third machine;
- receiving results of the query from the third machine; and
- displaying results of the query in a second window.
10. The method of claim 9, wherein said extracting step further comprises:
 - parsing the displayed data to locate text associated with an event, and
 - identifying text associated with the event.
11. The method of claim 9, wherein said composing step further comprises rebuilding the displayed data into a format for delivery to at least a third machine.
12. The method of claim 9, further comprising the steps of:
 - determining whether a page of the results contains a tag, and
 - displaying a button corresponding to the tag in the second window.
13. The method of claim 12, further comprising:
 - determining when the button is activated;
 - extracting the results of the query from the second window when the button is activated;
 - composing a booking transaction from the results; and
 - displaying a confirmation message in the first window.
14. A machine-readable storage medium having stored thereon machine executable instructions, the execution of said instructions adapted to implement a method for accessing information from multiple remote sources, said method comprising:
 - displaying data from a first machine in a first window;
 - extracting a portion of the displayed data from the first window;
 - composing a query from the extracted data;
 - transmitting the query from a second machine to at least a third machine;
 - receiving results of the query from the third machine; and
 - displaying results of the query in a second window.
15. The machine-readable storage medium of claim 14, wherein said extracting further comprises:
 - parsing the displayed data to locate text associated with an event, and
 - identifying text associated with the event.
16. The machine-readable storage medium of claim 14, wherein said composing further comprises rebuilding the displayed data into a format for delivery to at least a third machine.
17. The machine-readable storage medium of claim 14, further comprising:
 - determining whether a page of the results contains a tag, and
 - displaying a button corresponding to the tag in the second window.
18. A user interface for accessing information from multiple remote sources comprising:
 - a display for displaying at least two windows;
 - a first machine coupled to said display for:
 - extracting a portion of displayed data from a first window;
 - composing a query from the extracted data;
 - transmitting the query to at least a second machine; and
 - displaying results of the query in a second window; and
 - a second machine coupled to said display for:
 - determining whether a page of the results contains a tag, and

13

displaying a button corresponding to the tag in said second window;
determining when the button is activated;
extracting the results of the query from said second window when the button is activated;
composing a booking transaction from the results;
transmitting the booking transaction to the first machine; and
displaying a confirmation message in the first window.

19. A computer-implemented method for accessing information from multiple remote sources, comprising:

displaying data, from a first machine, in a first window on a display;

extracting a portion of displayed data from the first window;

14

composing a query from the extracted data;
transmitting the query to at least a second machine;
displaying results of the query in a second window on the display;

determining whether a page of the results contains a tag, and displaying a button corresponding to the tag in the second window;

determining when the button is activated;

extracting the results of the query from the second window when the button is activated;

composing a booking transaction from the results; and
displaying a confirmation message in the first window.

* * * * *